Joint Power Optimization of Data Center Network and Servers with Correlation Analysis

Kuangyu Zheng, Xiaodong Wang, Li Li, and Xiaorui Wang
The Ohio State University, USA
{*zheng.722, wang.3570, li.2251, wang.3596*}@*osu.edu*

***Abstract*—Data center power optimization has recently received a great deal of research attention. For example, server consolidation has been demonstrated as one of the most effective energy saving methodologies. Likewise, traffic consolidation has also been recently proposed to save energy for data center networks (DCNs). However, current research on data center power optimization focuses on servers and DCN separately. As a result, the optimization results are often inferior, because server consolidation without considering the DCN may cause traffic congestion and thus degraded network performance. On the other hand, server consolidation may change the DCN topology, allowing new opportunities for energy savings.**

In this paper, we propose PowerNetS, a power optimization strategy that leverages workload correlation analysis to jointly minimize the total power consumption of servers and the DCN. The design of PowerNetS is based on the key observations that the workloads of different servers and DCN traffic flows do not peak at exactly the same time. Thus, more energy savings can be achieved if the workload correlations are considered in server and traffic consolidations. In addition, PowerNetS considers the DCN topology during server consolidation, which leads to less inter-server traffic and thus more energy savings and shorter network delays. We implement PowerNetS on a hardware testbed composed of 10 virtual switches configured with a production 48-port OpenFlow switch and 6 servers. Our empirical results with Wikipedia, Yahoo!, and IBM traces demonstrate that PowerNetS can save up to 51.6% of energy for a data center. PowerNetS also outperforms two state-of-the-art baselines by 44.3% and 15.8% on energy savings, respectively. Our simulation results with 72 switches and 122 servers also show the superior energy efficiency of PowerNetS over the baselines.

I. INTRODUCTION

Recent years have seen a dramatic growth of large-scale Internet data centers due to the increasing popularity of clouding computing. Data centers are well known for their significant energy consumption, which is estimated to consume approximately $27 billion a year, and the number is anticipated to double by 2014 [1]. Many studies [2][3][4][5] show that there are typically three major power consumers in a data center: servers, cooling systems, and the data center network (DCN). Although the detailed power breakdown of a data center may vary from site to site, the approximate percentage of each power consumer is commonly estimated as: servers (40-55%), cooling systems (15-30%), and DCN (10-25%). Among the three, the power effieicency of data center cooling has been siginficantly improved in the recent years. For example, Google has lowered the cooling power to less than 6% through heavily customized optimization (e.g., using cold river water or outside cool air for cooling) [6]. Therefore, it is foreseeable that servers and networks are becoming the two most significant power consumption contributors in the future. Particularly, the percentage of network power can even increase to 50% in future data centers, where servers become more power-proportional to their workloads [7].

Data centers are generally provisioned based on a worst-case scenario, which often leads to the average server utilization being lower than 10% [8]. To improve the server power efficiency, server consolidation based on Virtual Machine (VM) migration has been demonstrated as one of the most effective energy saving methodologies [9][10][11]. In this approach, VMs are migrated and consolidated onto a small number of servers with respect to various performance and resource constraints (e.g., CPU utilization, memory capacity), such that unused servers can be turned off to save power. Similar to servers, a DCN is also commonly provisioned for the worst-case workloads that rarely occur. As a result, the capacity of a DCN can often become significantly underutilized. Therefore, dynamic traffic consolidation [4][12] has also been recently proposed to consolidate DCN traffic flows onto a small set of links and switches, such that unused network devices can be shut down for power savings[1]. As a result, DCNs can become more power-proportional to their workloads as well. However, current research on data center power optimization focuses on servers and DCN in a separate manner. As a result, the optimization results are often inferior, because server consolidation without considering the DCN may cause traffic congestion and thus degraded network performance. On the other hand, server consolidation may constantly change the DCN topology, allowing new opportunities for energy savings.

In addition, existing research on server and traffic consolidation commonly consolidates servers and traffic flows, respectively, based on a key assumption that the CPU utilization of each server or the bandwidth demand of each network flow can be approximated as a constant during the consolidation period. This is in contrast to the fact that the utilization of a VM or the bandwidth demand of a flow can vary over time. The variations can be significant because the consolidation period

---

This work was supported, in part, by NSF under grant CNS-1143607 (CAREER Award) and by ONR under grant N00014-09-1-0750.

978-1-4799-3360-0/14/$31.00

[1] Similar to related work [4][7][12], we use power and energy interchangeably here, because data center is typically required to be always on.

normally cannot be too short due to overheads considerations [10]. Therefore, existing work has to use either estimated maximum or average values to perform consolidation, which can result in either unnecessarily high power consumption or undesired server/link capacity violations, respectively. Our analysis of real data center traces (in Section IV) show that the utilizations of different servers or the bandwidth demands of different flows usually do not peak at exactly the same time. Furthermore, our analyses demonstrate that the 90-percentile utilizations or bandwidth demands are usually 60% or less of the peak values. Therefore, if we avoid consolidating servers or traffic flows that are positively correlated (i.e., peak at the same time) based on 90-percentiles instead of the peak values, we may save more energy during server and traffic consolidations.

In this paper, we propose PowerNetS, a power optimization strategy that leverages workload correlation analysis to jointly minimize the total power consumption of servers and the DCN in a data center. The design of PowerNetS is based on the key observations that more energy savings can be achieved if the workload correlations are considered in server and traffic consolidations. During a consolidation process, PowerNetS first analyzes the workload correlations among different VMs, as well as the traffic correlations among different flows. Based on the correlation analysis, PowerNetS consolidates VMs that are not positively correlated onto the same physical server, subject to the resource constraints of the servers. In the meantime, PowerNetS considers the DCN topology changes resulting from server consolidation and tries to consolidate VMs that are linked together through traffic flows onto the same server or servers close to each other. As a result, the DCN can have less inter-server traffic and thus more energy savings and shorter network delays. Specifically, this paper makes the following contributions:

- We propose to jointly optimize the power consumption of servers and the DCN in a data center, as the two are becoming the two most significant power consumers in the future. We identify that server consolidation without considering the DCN may cause traffic congestion and thus degraded network performance. On the other hand, server consolidation may change the DCN topology, allowing new opportunities for energy savings. Therefore, as shown in our evaluation, a naive combination of server and traffic consolidations leads to unnecessary energy wastes and undesired longer network delays.
- We analyze real data center traces, including 5,415 server traces from IBM and DCN traces from Yahoo! and Wikipedia. We observe that different VMs and network flows in data centers usually have weak correlations with each other and do not peak simultaneously. Based on the observations, we propose to consolidate servers and traffic flows based on their correlations, such that more power savings can be achieved.
- We design PowerNetS, a power optimization strategy that leverages workload correlation analysis to jointly minimize the total power consumption of servers and the DCN in a data center. We first mathematically formulate joint power optimization as a constrained optimization problem. We then propose a heuristic algorithm to find the consolidation solution with much lower overheads. Our results show that the energy saving and performance differences between PowerNetS and the optimal solution are sufficiently small.
- We implement PowerNetS both on a hardware testbed with a production 48-port OpenFlow switch and 6 servers, as well as in a larger-scale simulation with 72 switches and 122 servers. Both the empirical and simulation results with real traces demonstrate that PowerNetS outperforms two state-of-the-art baselines ([10] and [12]) by 15.8% and 44.3% on energy savings, respectively.

The rest of the paper is organized as follows. Section II reviews the related work. Section III introduces the formulation of the joint power optimization problem. Section IV describes the design of PowerNetS in detail with examples. Sections V and VI present the evaluation results on the hardware testbed and the simulation experiments, respectively. Finally, Section VII concludes the paper.

II. RELATED WORK

Many recent studies have proposed to reduce server energy consumption in data centers using server virtualization and VM consolidation [9][10][11][13][14]. Verma et al. propose pMapper [9] and CBP [10], which consolidate VMs into a small number of servers and then turn off unused servers for power savings. Similar to PowerNetS, CBP also considers correlation in the VM consolidation process but focuses only on the server side. In sharp contrast, PowerNetS analyzes the correlations of both server and DCN workloads to jointly optimize the total power of both sides in a coordinated way. As shown in our evaluation, such a joint optimization outperforms CBP by 15.8% in energy savings.

DCN power optimization has also received some attention recently. ElasticTree [4] consolidates different traffic flows onto a small set of switches and links to save power by turning off unused switches. Link adaptation is used in [7] which lowers the link speed dynamically to save port energy. CARPO [12] utilizes both correlation-aware traffic consolidation and link adaptation to achieve more DCN energy savings. However, none of the above studies consider the network topology variations caused by VM migration.

Much fewer studies have been conducted to jointly consolidate servers and DCN for improved power savings and network performance. Existing work focuses on either the network part power savings [15][16], or traffic bandwidth guarantees in VM placement [17][18]. The work that is most closely related to PowerNetS is [19], which converts joint power optimization into a unified routing problem that can be solved with a single solution. PowerNetS differs from [19] significantly because it utilizes correlation analysis in joint consolidation, which leads to much better energy saving than correlation-unaware methods. In addition, PowerNetS features a well-designed heuristic algorithm that incrementally performs VM and traffic consolidations with much lower overheads. Moreover, PowerNetS optimizes the total power consumption while maintaining the desired network performance, which is tested with extensive hardware and simulation experiments using real data center traces.

III. PROBLEM FORMULATION

In this section, we first introduce the server and DCN power models. We then formulate the joint server and DCN power optimization as a constrained optimization problem.

A. Power Models

We have the following notations for the power models of the servers and switches in the data center:

- $VM_{i,j}$: The jth virtual machine on the ith server;
- $u(VM_{i,j})$: Normalized CPU utilization of $VM_{i,j}$ on server i (e.g., $u(VM_{i,j})$ = 0.4 means $VM_{i,j}$ is using 40% CPU utilization of server i);
- P_i^{idle} : Idle power of server i;
- $P_i^{dynamic}$: Maximum dynamic power of server i;
- w_i : The number of VMs assigned to server i;
- $P_k^{chassis}$: The chassis power consumption of a switch k;
- $P_{l,k}^{port}$: The power consumption of port l on switch k;
- N : The total number of VMs;
- M : The total number of servers;
- K : The total number of switches;
- V : The total number of links in the data center;
- f_i : The total number of flows on $link_i$;
- $d_f^{link_i}$: The data rate of flow f on $link_i$;

To calculate the power consumption of server i, we adopt the power model based on CPU utilization [20] as follows.

$$P_i^{server}=\begin{cases} P_i^{idle}+\sum_{j=1}^{w_i} u(VM_{i,j})\times P_i^{dynamic} & , w_i>0; \\ 0 & , w_i=0. \end{cases} \tag{1}$$

P_i^{server} is composed of idle power and dynamic power. The idle power is considered as a constant while the dynamic power is linear to the total utilization of all the VMs on that server. When there is no VM assigned to the server, the server can be turned off to save power.

To obtain the power consumption of a switch k, we use the power model presented in [12], which is

$$P_k^{switch}=\begin{cases} P_k^{chassis}+\sum_{l=1}^{L_k} P_{l,k}^{port} & \text{if switch } k \text{ is on;} \\ 0 & \text{if switch } k \text{ is off.} \end{cases} \tag{2}$$

The total power of a switch is the sum of the chassis power and the port power. L_k is the number of active ports on switch k. If the switch is turned off, the power is then 0.

B. Problem Formulation

To jointly optimize the server and network power consumption periodically, with N virtual machines to be hosted on M servers and K switches in the data center, the problem in each period is formulated as:

$$\min(\textstyle\sum_{i=1}^{M} P_i^{server}+\sum_{k=1}^{K} P_k^{switch}) \tag{3}$$

subject to the following constraints:

$$\textstyle\sum_{i=1}^{M} w_i = N \tag{4}$$

$$\textstyle\sum_{j=1}^{w_i} u(VM_{i,j}) \le 1,\ \forall i\in[1,M] \tag{5}$$

$$\textstyle\sum_{f=1}^{f_i} d_f^{link_i} \le C^{link_i},\ \forall i\in[1,V] \tag{6}$$

Equations (4) and (5) are server-side constraints, which enforce that all the VMs must be hosted by a server and the utilization of all hosts must be less than 100% all the time. Equation (6) is the network-side constraint, which ensures that for any given $link_i$, the total data rate from all the f_i flows using that link must not exceed the link capacity C^{link_i}.

To solve the above optimization problem, optimization tools like MathProg can be used to provide a near-optimal solution. However, the computation complexity increases exponentially with the number of servers and switches [12]. In a typical data center with tens of thousands of severs and hundreds of switches, it is unrealistic to use such an optimization tool at this scale. To mitigate this problem, in the next section, we propose a system framework and a light-weight heuristic algorithm, which can achieve similar energy savings with the near-optimal solution but with much lower overheads.

IV. DESIGN OF POWERNETS

In this section, we first introduce the correlation analysis in the server and DCN traces. We then introduce PowerNetS, a joint server and network consolidation framework.

A. Correlation Analysis

Statistically, the correlation between two random variables, X and Y (x_i and y_i are respective sample points in a period), reflects their associated linearity, which can be quantified by the Pearson Correlation Coefficient below. Specifically, the more positive two variables are correlated, the more likely they will have their peak/valley values at the same time.

$$r_{XY}=\frac{n\sum x_i y_i-\sum x_i\sum y_i}{\sqrt{n\sum x_i^2-(\sum x_i)^2}\sqrt{n\sum y_i^2-(\sum y_i)^2}} \tag{7}$$

In the correlation-aware consolidation approach, different VMs (or DCN flows) are consolidated based on their non-peak values and inter-correlation relationship. This approach is based on two key observations from the real data center server and DCN workload. First, in majority of the time, the workload of a server (or a traffic flow) is much lower than its peak value. Figure 1 shows the cumulative distribution function (CDF) of 6 servers' workload traces randomly chosen from 5,415 server traces of an IBM data center [21]. We can see that for all the flows, their workloads are less than 60% of peak value for 90% of the time. The second observation is that the workloads of different servers (or traffic flows) are weakly correlated, which means they do not peak at exactly the same time. Figure 2 is the correlation matrix of 100 randomly chosen server traces from the IBM trace file, which shows the pairwise correlations of most server workloads are less than 0.5.

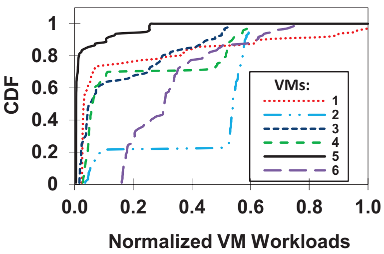

Fig. 1. CDF distribution of 6 randomly selected IBM server traces used in hardware experiments.

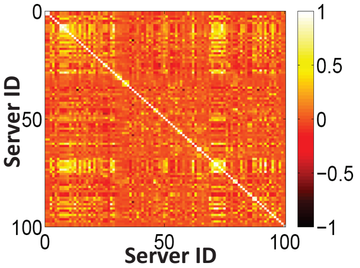

Fig. 2. Correlation coefficients between 100 randomly chosen IBM servers' workload traces.

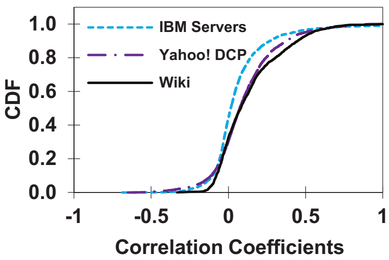

Fig. 3. CDF correlation distribution of Wikipedia, Yahoo! DCN traces and IBM server traces.

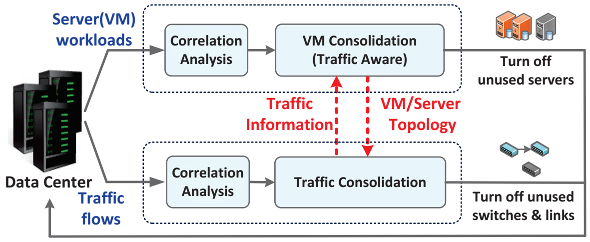

Fig. 4. The proposed framework of PowerNetS.

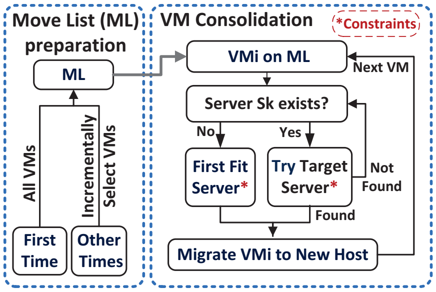

Fig. 5. Illustration of the VM consolidation coordinated with traffic consolidation. See Section IV-C for details, and Figures 6 and 7 for examples.

The same observations apply to the network traffic flows in the data center. In Figure 3, we plot the CDF of correlation coefficients between every two traffic flows of two DCN traces: Wikipedia [22] and Yahoo! [23]. We see that more than 90% of the pairwise correlations between every two traffic flows are less than 0.5. Based on these two observations, our correlation-aware approach consolidates different VMs (or traffic flows) using their non-peak workload (e.g., 90-percentile workload), under the condition that the correlation between these VMs (or traffic flows) is below a threshold. This approach can lead to more energy savings with the same amount of resource, while reducing the chance of capacity violation. Our observations are consistent with the correlation analyses performed in related work [10] and [12].

B. *PowerNetS Framework*

Figure 4 shows the framework of PowerNetS, which is composed of a VM consolidation module and a traffic consolidation module. The two modules periodically analyze the server workloads and traffic flows from the data center, respectively, and make the consolidation decisions in a coordinated way. The main feature of the VM consolidation module is its DCN topology awareness. First, it tries to place the VMs that are linked through a network flow onto the same server, or servers as close as possible, such that the required number of network hops can be minimized. Note that if the source and destination of a VM pair are placed onto the same server, the network flow between them becomes a with-in server flow, which does not requires network resources. This reduces the workloads of the DCN and provides more space for energy savings. Second, it calls the traffic consolidation module to check whether there is a physical path available for the flow to be assigned to and if the flow can be consolidated with any existing flows to save the DCN power. Upon called, the traffic consolidation module first checks the correlations between the flow to be assigned and the flows that have already been assigned to that physical path. It then consolidates traffic flows using the correlation-aware method based on the non-peak traffic load values. The traffic consolidation result is then given back to the VM consolidation module to update the link bandwidth condition of each server. More details are presented in the next section.

Because both server and DCN workloads can vary over time, it is important to perform consolidations periodically in response to any workload variations. However, it is also well known that VM migration and consolidation can introduce non-trivial overheads [9][24]. Hence, the selection of consolidation period should be a trade-off between system responsiveness and consolidation overheads. In this paper, we propose to run the traffic consolidation module at a smaller time interval (e.g., 10 minutes), while running the VM consolidation module at a larger interval (i.e., one day). To address the overheads of VM migration, the VM consolidation is also run in an incremental fashion. In each VM consolidation period, PowerNetS only incrementally selects a set of VMs from those overloaded servers (i.e., utilization > 100%) and the least utilized server as VM migration candidates. The rationale of selecting the least utilized server is that we can turn off those servers for power savings once their VMs are migrated to other servers. With this incremental VM selection method, the consolidation overheads can be significantly reduced and better amortized over the entire consolidation period.

C. *Algorithm of PowerNetS*

Figure 5 illustrates the details of the PowerNetS consolidation algorithm. At the beginning of each VM consolidation period, PowerNetS first prepares the moving list of VMs to be migrated based on the the workload analysis. If it is the initial time, all VMs are selected. Otherwise, it will apply incremental selection by 1) selecting the smallest number of VMs from those overloaded servers till all the servers are no longer overloaded, and 2) selecting all VMs from the server with least utilization.

When the moving list is ready, PowerNetS begins the VM consolidation of each VM based on their demands in a decreasing order. For a VM_i from list, the VM consolidation

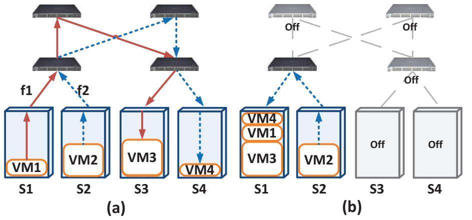

Fig. 6. PowerNetS example (initial consolidation) with 4 servers, 4 VMs and 2 flows. (a) Before consolidation. (b) After consolidation. (Moving List: VM_3 (60%), VM_2 (50%), VM_1 (25%), and VM_4 (15%))

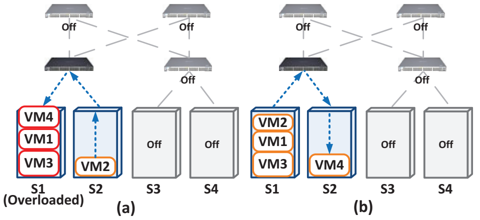

Fig. 7. PowerNetS example (incremental consolidation). (a) Before consolidation, server S_1 is overloaded (b) After consolidation. (Moving List: VM_4 (30%), VM_2 (20%))

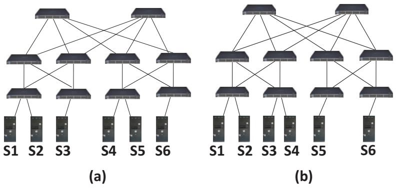

Fig. 8. The 6-server topologies used in the hardware experiments: (a) Symmetric topology with 3 servers on each side. (b) Asymmetric topology with 4 servers on the left side and 2 servers on the right side.

module first finds its pair VM that is linked with a flow, denoted as VM_j, and checks if it is already placed in a host server. If VM_j is not on any server (this means VM_j is also moved to the moving list), the current VM_i is then assigned a host based on the first-fit bin packing algorithm. In the consolidation, two server constraints must be met: (1) The resource requirement of the VM's demand should not exceed the current available resources of the candidate host (i.e., total CPU utilization demand of VMs should be less than the server capacity). (2) Correlation coefficient between the new VM and any existing VMs on the candidate host is below the correlation threshold.

On the other hand, if VM_j is already on a server S_k, S_k then becomes the target server for VM_i because PowerNetS with the traffic-aware feature tries to consolidate a source and destination pair of VMs into the same server to reduce the number of flows in the DCN. If both constraints (1) and (2) are met, VM_i will be migrated to S_k, the flow between VM_i and VM_j then becomes a within-server flow. But if one of the constraints is not satisfied, PowerNetS will search for another server with the shortest distance (i.e., smallest number of link hops) to S_k as the new target server. This time, in addition to constraints (1)(2), the VM consolidation module will resort to the traffic consolidation module with flow information to check two traffic constraints: (3) Possible bandwidth demand of the considered flow should not exceed the remaining link capacity of a candidate routing path, and (4) The correlation value between the considered flow from the new VM and any existing flows on a candidate path is below the traffic correlation threshold. This checking process continues until a candidate target host is found to meet all the constraints. If such a server is not found, PowerNetS will relax the network constraints by only considering (1)(2) to choose the first-fit server with the shortest distance to S_k.

D. Example

To make it clearer, we use a simple example with 4 servers, 4 VMs and 2 flows to demonstrate how PowerNetS consolidates servers and network traffic flows to save power.

As shown in Figure 6(a), the normalized 90-percentile value of CPU resource demand of each VM is set as: 25% (VM_1), 50% (VM_2), 60% (VM_3) and 15% (VM_4). Meanwhile, there are two flows: f_1 from VM_1 to VM_3 and f_2 from VM_2 to VM_4, with normalized 90-percentile bandwidth demands as 80% (f_1) and 50% (f_2). To demonstrate the correlation-aware consolidation feature, we set the correlation threshold to zero, and the correlation coefficient between VM_2 and VM_4 to +1 which is the only coefficient exceeds the threshold.

PowerNetS initially puts all the VMs into the moving list and the list is sorted by demand size in a decreasing order. VM_3 is the first VM chosen for placement as it is the first VM in the moving list. As a traffic-aware scheme, when choosing a suitable server to place VM_3, PowerNetS first checks if VM_1 is on any server since there is a flow between VM_3 and VM_1. Because VM_1 is in the moving list, VM_3 is then assigned to server S_1 according to the first-fit rule without violating any constraints. When placing the next VM, VM_2, it turns out server S_1 cannot be the host since only 40% of CPU resource is available. Thus VM_2 is placed in server S_2 without violating any constraint. VM_1 is the third VM to be placed. Note that VM_1 and VM_3 are the source and destination of flow f_1. With traffic awareness, PowerNetS first considers S_1 which is the host of VM_3. Without violating any constraints, VM_1 is assigned to S_1, therefore flow f_1 becomes a within-server flow, which reduces the workload of the network such that more DCN energy can be saved. The last VM to be placed is VM_4. Same as the process of assigning VM_1, since there is a flow f_2 between VM_2 and VM_4, PowerNetS first checks if VM_4 can be assigned to server S_2, which is the host of VM_2. However, this attempt fails because of the correlation constraint violation between VM_2 and VM_4. Therefore, PowerNetS tries the next closest server S_1 for VM_4. Since no constraints are violated, VM_4 is migrated to S_1. Then the unused servers and switches are turned off to save power. The consolidation results of the initial period are shown in Figure 6(b).

Figure 7 shows the incremental consolidation procedure of PowerNetS. At the beginning of a new period (Figure 7(a)), Server S_1 becomes overloaded due to changes of VMs resource demands: 35% (VM_1), 20% (VM_2), 40% (VM_3) and 30% (VM_4). According to the incremental method, VM_4, as the VM with smallest demand, is selected to address the overloaded situation. Because S_2 is the least utilized active server, VM_2 is also selected. Then the moving list has VM_4 (30%) and VM_2 (20%) to be placed. Following the same consolidation rules, VM_4 is placed on the first-fit server S_2. When placing VM_2, even S_2 is the first target server, because VM_4 and VM_2 can not be put together due to correlation violation, VM_2 is migrated to the nearest first-fit server S_1. The final consolidation result is shown in Figure 7(b).

V. HARDWARE EXPERIMENTS

In this section, we first introduce the testbed setup for the evaluation of PowerNetS and the baseline algorithms. We then present the results from different hardware experiments.

Fig. 9. Hardware testbed with 6 servers and 10 virtual switches configured with a production Pica8 48-port OpenFlow switch.

A. Testbed and Experimental Setup

The hardware testbed consists of six servers (running six VMs in total) and a 48-port OpenFlow-enabled Pica8 3290 switch. Each server is equipped with one Intel Core2 Duo E4600 2.40GHz processor and 2GB RAM. All servers use Ubuntu 12.04 with Xen 4.1.2 as the environment to host and control the six VMs. Meanwhile, to build the network topologies shown in Figure 8, we use the same method in [12] to divide the 48-port OpenFlow switch into 10 four-port virtual switches. The OpenFlow switch is connected to a controller computer other than the six servers. With the information of workloads and traffic condition, the controller periodically calculates the assignment of VM and the traffic flow dynamically changes the network routing paths and VM locations by VM migration.

As discussed in Section IV, PowerNetS uses a longer time interval as the VM consolidation period and a shorter interval for traffic consolidation to reduce the overheads cost of frequent VM migration. In the following experiments, we set the VM consolidation period as one day and the traffic consolidation interval as 10 minutes, which are the same settings used in [10] and [12], respectively, for fair comparison. Likewise, 90-percentile workload values are used for consolidation and the correlation thresholds are set to 0.3, in both the VM consolidation and traffic consolidation. Note that these settings are user-defined and can be adjusted according to characteristics of different data center workloads and performance requirements.

In all the experiments, each VM is controlled to follow the real CPU utilization of a server randomly selected from the 5,415 IBM's data center servers [21]. The network flow between each VM pair is also from real data center network traces: the Wikipedia traces [22] and the Yahoo! data center traces [23]. For each network flow, two VMs are picked, one as source and the other one as destination. We measure the power consumption of the hardware testbed, as well as the average packet delay of each flow. The power measurement is from the power meter connected to all the servers and the physical switch. Since we cannot measure the power consumption of a virtual switch individually as they all belong to the same physical switch, we use the same method as in [12] to calculate the power of the virtual switch as follows. The total idle power of the physical switch in the testbed is 66.3W, so the idle power for each of the 10 virtual switches is evenly treated as 6.63W. If one unused virtual switch can be turned off after traffic consolidation, 6.63W is subtracted from the measured switch power consumption. Note that we turn off the unused ports on the OpenFlow switch such that the port power consumption is already included in the power measurement.

We compare PowerNetS with the following baselines:

- *CBP* [10] is a correlation-aware VM consolidation scheme. It consolidates VMs onto physical servers based on their inter-correlation and non-peak CPU utilization. Different from PowerNetS, CBP does not consider the traffic and network topology during consolidation. In the experiment CBP picks one available path randomly for each flow, and keeps all links and switches active.
- *CARPO* [12] is a state-of-the-art network power saving strategy which consolidates traffic flows onto a sub-set of the DCN, based on correlation and non-peak workload of different traffic flows. Different from PowerNetS, it does not involve VM consolidation for server power saving. Therefore all servers must be kept active, with one VM on each server.
- *CBP+CARPO* is a simple combination of CBP and CARPO. Because CBP decides the locations of VMs that determine the sources and destinations of traffic flows, this method first executes CBP to consolidate VMs with the long period, then applies CARPO to adjust traffic paths and turn on/off switches with the short period. Note that, due to traffic-unaware VM consolidation, network congestion can be incurred even with traffic adjustment.
- *Non_corr* is a variant of PowerNetS that is similar to [19]. It consolidates VMs and traffic flows based on peak workload values without any consideration of correlation.
- *PowerNetS_all* is another variant of PowerNetS. Different from the incremental strategy used by PowerNetS which only selects VMs from overloaded servers and the least utilized servers, PowerNetS_all conducts complete re-consolidation that reconsiders and moves all the VMs in each period. This method has much larger overheads compared with PowerNetS.
- *Optimal* is the optimal solution of the joint power optimization problem that formulated in Section IV. This solution is derived from an exhaustive search approach. Due to its high computation complexity, we only test it in the 6-server hardware experiments.

B. Results with Wikipedia DCN Traces

In this set of experiments, the 7-day Wikipedia DCN traces are used as the network workloads. We evaluate the energy saving ratio of different schemes, compared with the original scenario (i.e., no power management). A balanced network topology shown in Figure 8(a) is used in the testbed.

Figure 10 shows the energy saving results of different power management schemes. Optimal can save the most amount of energy (55.1% on average) compared to the energy consumption without power management. PowerNetS comes second with its energy savings just slightly less than that of Optimal. It saves 51.6% of energy and has higher savings than the other three baselines on each day. In contrast, CARPO and CBP only save 7.3% and 35.8% of the total energy, respectively, which are lower than that of PowerNetS by 44.3% (=51.6%-7.3%) and 15.8% (=51.6%-35.8%). This is because

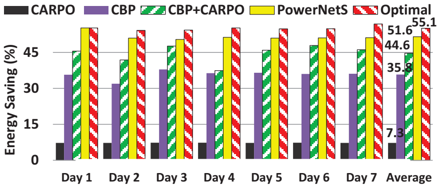

Fig. 10. Energy saving using the Wikipedia DCN traces with the topology in Figure 8(a).

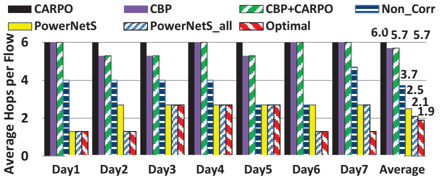

Fig. 11. Average number of hops per flow using the Wikipedia DCN traces with the topology in Figure 8(a).

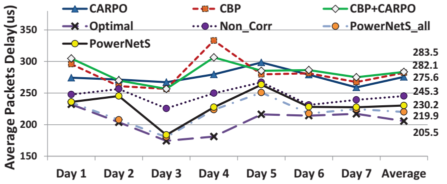

Fig. 12. Average packet delay using the Wikipedia DCN traces with the topology in Figure 8(a). The average result is shown at the end of each line.

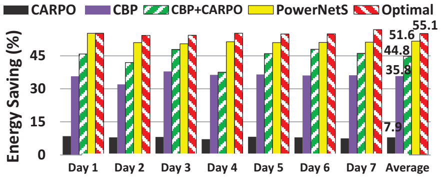

Fig. 13. Energy saving using the Wikipedia DCN traces with the topology in Figure 8(b).

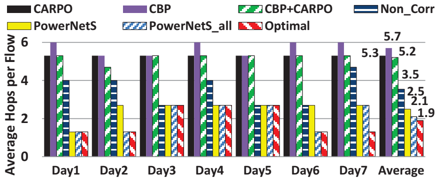

Fig. 14. Average number of hops per flow using the Wikipedia DCN traces with the topology in Figure 8(b).

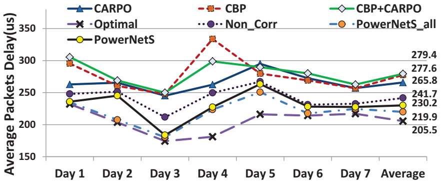

Fig. 15. Average packet delay using the Wikipedia DCN traces with the topology in Figure 8(b). The average result is shown at the end of each line.

CARPO or CBP saves either the networking or server energy without considering them together. The simple combination, CBP+CARPO saves around 44.6% energy on average, which is worse than PowerNetS by 7.0% on energy savings. This is because directly adding CARPO on the top of CBP cannot lead to a good network energy saving result, as CBP consolidates VMs without considering the impact on network, which may result in smaller consolidation space in DCN for CARPO.

Figure 16 shows the energy savings of PowerNetS and its two variant schemes. We see that Non_corr, the scheme using peak workload values for consolidation, only achieves 28.9% energy savings, which is 22.7% less than PowerNetS that uses non-peak workload in correlation analysis. This illustrates the advantage of PowerNetS to use non-peak percentile values in power management. The other variant scheme PowerNetS_all applies complete consolidation with all the VMs in each period. Compared with the incremental VM consolidation method of PowerNetS, it achieves a little higher energy savings (0.9%) but has much more overheads with all 6 VMs migrated. In contrast, PowerNetS moves only 2.14 VMs on average. Therefore PowerNetS is a better choice with less overheads but almost the same amount of energy savings.

To evaluate the network performance of different schemes, we measure the average packet delay in all the experiments. We also evaluate the average hops per flow, as it is one of the contributors to the end-to-end packet delay.

The results of average delay and average number of hops per flow are shown in Figure 11 and 12, respectively. Optimal has the least average delay as 205.5 μs, with 1.9 hops for each flow on average. Following Optimal, PowerNetS uses 2.5 hops per flow on average with 230.2 μs average delay. One interesting result is although CARPO uses 6 hops on average, which is more than the 5.7 hops in both CBP and CBP+CARPO, it has less average delay. This is because without considering the impact on network performance, the VM consolidation of CBP may lead to more flow congestions, and thus a longer delay. CBP+CARPO can adjust paths with traffic consolidation. It can help to reduce traffic congestion by choosing new link paths (e.g., on Day 4), but works only on links between switches, because the links between a server and a switch is determined by CBP and so unchangeable for CARPO.

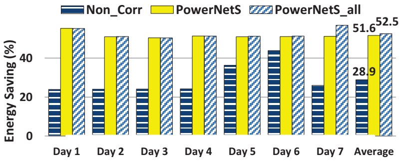

Fig. 16. Energy saving of the PowerNetS variants in the testbed experiments with the topology in Figure 8(a).

In addition, among different PowerNetS variants, Non_corr has longer flow distances (3.7 hops/flow) than PowerNetS due to using more switches and links, which leads to a longer delay. Compared with PowerNetS, PowerNetS_all has two more days achieved the optimal setting, which results in fewer average hops and a shorter delay. However, this comes at the cost of higher VM migration overheads than PowerNetS.

C. Results with Different Topology

In this set of experiments, we change the symmetric topology used in the previous experiments (Figure 8(a)) to an asymmetric topology as shown in Figure 8(b).

From the energy saving comparison in Figure 13, we can see that in the new asymmetric topology, the energy savings of CARPO increases from 7.3% to 7.9% on average, and CBP+CARPO also has an improvement from 44.6% to 44.8%. This is due to the fact that in the asymmetric topology, servers have a shorter distance between each other, and thus the traffic flows in CARPO have fewer hops. Meanwhile, Optimal, PowerNetS and CBP have the same energy consumption as in the symmetric topology, since the consolidation results turn out to be equal with both topologies here. As expected, PowerNetS still has the energy savings closest to Optimal.

To compare the network performance in the asymmetric topology, we show the average number of hops per flow of each scheme in Figure 14 and the average network delay in Figure 15. Due to the change of topology, the average hops per flow in CARPO has reduced to 5.3, while those of Optimal and PowerNetS remain the same. Accordingly, as shown in Figure 15, CARPO and CBP+CARPO achieve shorter average delays compared with the results of using the symmetric topology. Nevertheless, compared with CARPO, CBP and CBP+CARPO, PowerNetS still has the smaller number of

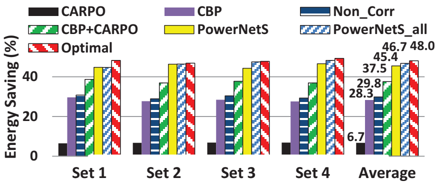

Fig. 17. Energy Saving using the Yahoo! DCN traces with the topology in Figure 8(a).

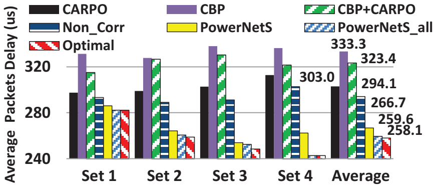

Fig. 18. Average packet delay using the Yahoo! DCN traces with the topology in Figure 8(a).

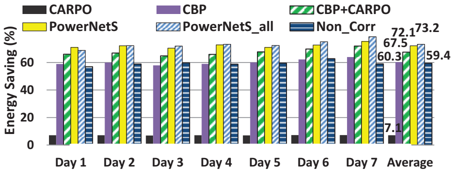

Fig. 19. Energy saving comparison for 7 days in a larger-size network with 122 servers.

average hops per flow and the shortest delay on average, due to its traffic-aware consolidation property.

D. Results with Yahoo! DCN Traces

We now evaluate PowerNetS using Yahoo! DCP data center traces [23], which have higher average data rates than the Wikipedia traces. Because the Yahoo! traces contain only data for a length of a single day, we repeat the 1-day experiment with 4 different sets of flows. For each set, we randomly choose 3 different flows from the total 70 flow traces to assign to the 3 pairs of source and destination VMs.

Figure 17 shows the energy savings of different power management schemes. We see that, similar to the Wikipedia results, PowerNetS can save as much as 45.4% of energy, which also outperforms the baselines CARPO, CBP, CBP+CARPO and Non_corr by 38.7%, 17.1%, 7.9% and 15.6% respectively. Meanwhile, it only has 2.6% less savings than Optimal.

Figure 18 shows the delay performance of all schemes. Compared with the previous experiments using Wikipedia DCN files, we see that due to the increase of traffic loads of Yahoo! traces, the average delays of different methods all increase. PowerNetS still has a low average delay of 266.7us, which is about 14.3% lower than that of CARPO, 20.0% lower than CBP, and 17.5% lower than CBP+CARPO. Meanwhile, PowerNetS has an average delay that is only 7.1μs longer than PowerNetS_all and 8.6μs longer than Optimal, with much lower complexity and overheads.

VI. SIMULATION EXPERIMENTS

In this section, we conduct simulation experiments on a larger scale topology, consisting of 122 servers and 72 switches. The network topology used in this experiment is a fat-tree topology [25]. Note that the design of PowerNetS does not rely on particular network structures.

A. Experimental Setup

For the simulation we assume all the servers are homogeneous. Each server is set to have 100W of idle power and 100W of maximum dynamic power based on the data from [20]. We use 122 VMs in total in the experiments, one on each physical server at the initial stage. Each VM runs one workload from the IBM CPU trace, and 61 Wikipedia network flow trace are used between the 61 pairs of VMs. The switches used in the simulation are 8-port switches. The chassis power of each switch is set to 10W, while the port power is set to 1W per port in the simulation according to [12].

B. Energy Savings

From Table I we can find PowerNetS on average uses more servers than CBP. This is because PowerNetS needs to consider network traffic condition in VM consolidation, such that it may use more servers to avoid traffic congestion. On the other hand, since PowerNetS can consolidate more traffic to reduce network power consumption, PowerNetS can still have lower total power consumption.

TABLE I

THE AVERAGE NUMBER OF ACTIVE DEVICES IN 122-SERVER SIMULATION. MORE ACTIVE DEVICES RESULT IN HIGHER POWER CONSUMPTION.

	# of Active Servers	# of Active Switches	Total Power (W)
CARPO	122	41.9	15,701.0
CBP	26.1	72.0	6,720.7
CARPO+CBP	26.1	45.9	5,490.6
PowerNetS	28.0	13.3	4,709.9
PowerNetS_all	26.6	12.9	4,533.1
Non_corr	37.3	18.4	6,871.2

Figure 19 is the power consumption results from the simulation. It shows that PowerNetS and PowerNetS_all always save more amount of power during the experiments. PowerNetS saves about 72.1% of the total power on average compared with the power consumption with no power mangement scheme. CARPO merely saves 7.1% of the total power since it only saves the networking energy consumption without considering the server energy consumption. Compared with the energy savings from CBP, PowerNetS outperforms it by 11.8% as CBP saves only 60.3% on average. This is because CBP is a traffic-unaware VM consolidation method, whose only goal is to save the server power without considering the network power consumption. Meanwhile, PowerNetS has 4.6% more power savings than CBP+CARPO. However, since the CBP+CARPO does not consider the impact on network traffic when consolidating VM, it risks to have more network congestion which degrades network performance.

C. Network Performance

To evaluate the network performance when the data center is operating under different power management schemes, we use OPNET [26] to build the 122-server network model, and run the experiment with real data center traces.

We first define a metric called “within-server flow ratio”, which is the number of flows within a single server divided by the total number of flows in the DCN. This metric accounts for how many network flows have been consolidated into a single server, such that the network workload is actually reduced. Figure 20 shows that PowerNetS achieves the second best within-server flow ratio (next only to PowerNetS_all that has much higher migration overheads). This means that PowerNetS successfully consolidates more VMs that are linked with network flows onto the same machine and thus reduces the networking workload. Figure 21 shows the average number of hops after consolidation by each power management scheme. We see that PowerNetS also has a smaller number of hops per flow than CARPO, CBP, CBP+CARPO and Non_corr. The results demonstrate that PowerNetS can significantly reduce

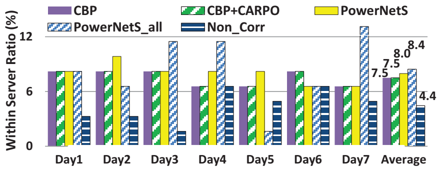

Fig. 20. Within-server flow ratio in the 122-server simulation (CARPO is now shown because it has no within-server flows).

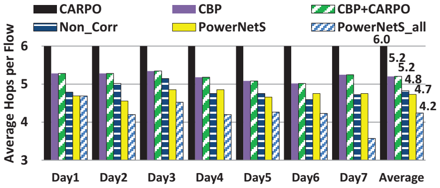

Fig. 21. Average hops per flow in the 122-server simulation using the Wikipedia DCN traces.

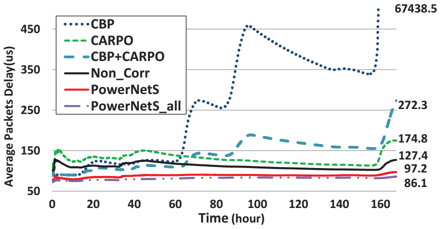

Fig. 22. The seven-day variation of average packet delay in the 122-server simulation (overall average delay is shown at the end of each line).

the average number of hops for each flow, which can in turn lead to shorter delays and a smaller chance for packets to get dropped in the network.

Figure 22 demonstrates the dynamic delay variation of the network flows in the seven days. For each point in this figure, it represents the average delay value since the beginning of the experiment. Except PowerNetS_all, PowerNetS always has the shortest delay compared with the other methods, no matter how the traffic loads change. However, for CBP and CBP+CARPO, when traffic loads increase at 63 hours, both of them encounter a much longer delay. When the traffic loads become much heavier at around 160 hours, congestion in CBP causes its delay dramatically increased (data higher than 600us is not plotted). This result directly demonstrates the importance of jointly considering VM and DCN in consolidation. Similar as in the small scale experiments, PowerNetS_all achieves the shortest delay at the cost of migrating all 122 VMs in every period. In contrast, PowerNetS has much less overheads (40.4 VMs) and outperforms all the other baselines.

Compared with the small scale testbed results in Section V, as topology scales up, there are more servers connected to the same edge switch, which provides PowerNetS more chances to put VMs pairs that are linked by flows in a near distance. Moreover, in a larger topology, as the number of switches and servers increases, there is also a greater energy saving potential for PowerNetS.

VII. CONCLUSION

In this paper, we have proposed PowerNetS, a power optimization strategy that leverages workload correlation analysis to jointly minimize the total power consumption of servers and the DCN in a data center. The design of PowerNetS is based on the key observations that the workloads of different servers and DCN traffic flows do not peak at exactly the same time. Thus, more energy savings can be achieved if the workload correlations are considered in server and traffic consolidations. In addition, different from existing work that consolidates servers and the DCN flows in a separate manner, PowerNetS considers the DCN topology changes during server consolidation, which leads to less inter-server traffic and thus more energy savings and shorter network delays. We first mathematically formulate joint power optimization as a constrained optimization problem. We then present a heuristic algorithm to find the consolidation solution that migrates the VMs in an incremental manner for much lower overheads. PowerNetS is implemented and evaluated on a hardware testbed composed of 10 virtual switches configured with a production 48-port OpenFlow switch and 6 servers. Our empirical results with Wikipedia, Yahoo!, and IBM traces demonstrate that PowerNetS can save up to 51.6% of energy for a data center. PowerNetS also outperforms two state-of-the-art baselines, CBP [10] and CARPO [12], by 15.8% and 44.3% on energy savings, respectively. Our simulation results with 72 switches and 122 servers also show the superior energy efficiency of PowerNetS over the baselines.

REFERENCES

[1] N. Knupffer, “The efficient datacenter,” 2011. [Online]. Available: http://blogs.intel.com/technology/2011/10/the_efficient_datacenter_-_not/
[2] United States Environmental Protection Agency, “Report to congress on server and data center energy efficiency,” 2007.
[3] A. Greenberg *et al.*, “The cost of a cloud: Research problems in data center networks,” *SIGCOMM CCR.*, vol. 39, no. 1, Dec. 2008.
[4] B. Heller *et al.*, “ElasticTree: Saving energy in data center networks,” in *NSDI*, 2010.
[5] S. Pelley *et al.*, “Understanding and abstracting total data center power,” in *WEED*, 2009.
[6] “Data center efficiency: How we do it,” 2012. [Online]. Available: http://www.google.com/about/datacenters/efficiency/internal/
[7] D. Abts *et al.*, “Energy proportional datacenter networks,” in *ISCA*, 2010.
[8] P. Padala *et al.*, “Adaptive control of virtualized resources in utility computing environments,” in *EuroSys*, 2007.
[9] A. Verma *et al.*, “pMapper: Power and migration cost aware application placement in virtualized systems,” in *Middleware*, 2008.
[10] ——, “Server workload analysis for power minimization using consolidation,” in *USENIX*, 2009.
[11] Y. Wang *et al.*, “Power optimization with performance assurance for multi-tier applications in virtualized data centers,” in *ICPPW*, 2010.
[12] X. Wang *et al.*, “CARPO: Correlation-aware power optimization in data center networks.” in *INFOCOM*, 2012.
[13] R. Nathuji *et al.*, “VirtualPower: Coordinated power management in virtualized enterprise systems,” in *SOSP*, 2007.
[14] W. Jiang *et al.*, “Joint VM placement and routing for data center traffic engineering,” in *INFOCOM*, 2012.
[15] V. Mann *et al.*, “VMFlow: Leveraging VM mobility to reduce network power costs in data centers,” in *IFIP Networking*, 2011.
[16] L. Wang *et al.*, “Joint virtual machine assignment and traffic engineering for green data center networks,” in *Greenmetrics*, 2013.
[17] Z. Jing *et al.*, “Towards bandwidth guarantee in multi-tenancy cloud computing networks,” in *ICNP*, 2012.
[18] X. Di *et al.*, “The only constant is change: Incorporating time-varying network reservations in data centers,” in *SIGCOMM*, 2012.
[19] H. Jin *et al.*, “Joint host-network optimization for energy-efficient data center networking,” in *IPDPS*, 2013.
[20] C. Lefurgy *et al.*, “Server-level power control,” in *ICAC*, 2007.
[21] X. Wang *et al.*, “SHIP: Scalable hierarchical power control for large-scale data centers,” in *PACT*, 2009.
[22] G. Urdaneta *et al.*, “Wikipedia workload analysis for decentralized hosting,” *Elsevier Computer Networks*, 2009.
[23] Y. Chen *et al.*, “A first look at inter-data center traffic characteristics via yahoo! datasets,” in *INFOCOM*, 2011.
[24] F. Travostino, “Seamless live migration of virtual machines over the MAN/WAN,” in *Supercomputing*, 2006.
[25] M. Al-Fares *et al.*, “A scalable, commodity data center network architecture,” in *SIGCOMM*, 2008.
[26] OPNET Technologies, Inc. [Online]. Available: http://www.opnet.com